

Static and Method Overloading

static

- One per class, not per object
- static variables
 - all instances of the class share the same static variable
 - example: serial/VIN number for Car
- static methods
 - invoked through class name, not object name
 - cannot reference instance variables
 - example: Math methods, isleap, ConvertTemperature

Method Overloading

- Method signature – name of method and type and order of parameters
- Two methods must have different signatures, but may have the same name
 - `println(String s)`
 - `println(int i)`
- A class may have multiple constructors with different parameter lists

Examples

- Student
 - Constructor 1 takes first name, last name
 - Constructor 2 takes first name, last name, grade
- StudentDB
 - addStudent(Student s)
 - addStudent(String fname, String lname)

Aggregation

- “Has-a” relationship
- An object can have references to other objects
 - Example:
 - ContactInfo – Name, Address, Phone
 - Employee “has-a” ContactInfo

Exercises

- Implement a class LoginGenerator. The class should have two static methods, both named generateLogin. The first method will take as input two Strings representing the first name and last name and return a String containing the first character of the first name, the first five characters of the last name, and a random number in the range 10-99. The second method will take as input one String representing the last name and will return a String containing the first five characters of the last name and a random number in the range 10-99. Make sure to handle the possible error that would occur if the last name were shorter than 5 characters.

Exercises

- Design and implement a class `Course` that contains information about a course taken at a school (for example, the course number, name, and instructor). Next, implement a class `Student` that contains the name of a student, the user login for the student, and up to four courses for which the student is registered. The `Student` constructor should take the first and last name of the student and use the `GenerateLogin` class to generate the Student's user login. Provide appropriate accessor and modifier methods for the `Student` class.